



DAT315

SQL Server 2005: VLDB Availability and Recovery Strategies

Kimberly L. Tripp
President, SQLskills.com

Speaker – Kimberly L. Tripp

- Independent Consultant/Trainer/Speaker/Writer

- Founder, SYSolutions, Inc. www.SQLskills.com

email: Kimberly@SQLskills.com

Become a subscriber on SQLskills.com and learn about new resources which can improve your productivity and server performance!

- Microsoft Regional Director <http://www.microsoftregionaldirectors.com/Public/>
- SQL Server MVP <http://mvp.support.microsoft.com/>
- Author for some SQL Server 2005 Whitepapers on MSDN (links from www.SQLskills.com)
- Coauthor MSPress: SQL Server 2000 High Availability, Presenter/Technical Manager for SQL Server 2000 High Availability DVD
- Writer/Editor for SQL Magazine www.sqlmag.com

Overview

- Resource and Database Availability
- Structured Database Design
 - Filegroups
 - Partitioning
- Availability with Damaged Devices
- Piecemeal Backup/Restore with Minimized Downtime
- Database Snapshots
 - Creation
 - Usage in Recovery from Human Error

If You Could...

- Control a database at a finer granularity (than the database level) would it offer better availability?
- Heavily dependent on application (and the need for the data that's not available)
- Depends on the database structures:
 - Table/Index Structures
 - Locking

What Happens When...

- Isolated hardware failure
 - A hard drive crashes
 - A page is damaged within a database
- Human error
 - An administrator performs an incorrect modification
 - A user/application performs an incorrect or even possibly a malicious modification
- In SQL Server 2000
- In SQL Server 2005

In SQL Server 2000

- Isolated hardware failure
 - Entire Database is offline/inaccessible
 - Recovery—even if partial—needs to be rolled forward completely using transaction log backups
- Human error
 - Need to determine if entire database should be taken offline
 - Recover database to earlier point in time through proper (and time consuming) restore sequence
 - Restore to alternate location and manually merge in data (time consuming/error prone)

In SQL Server 2005

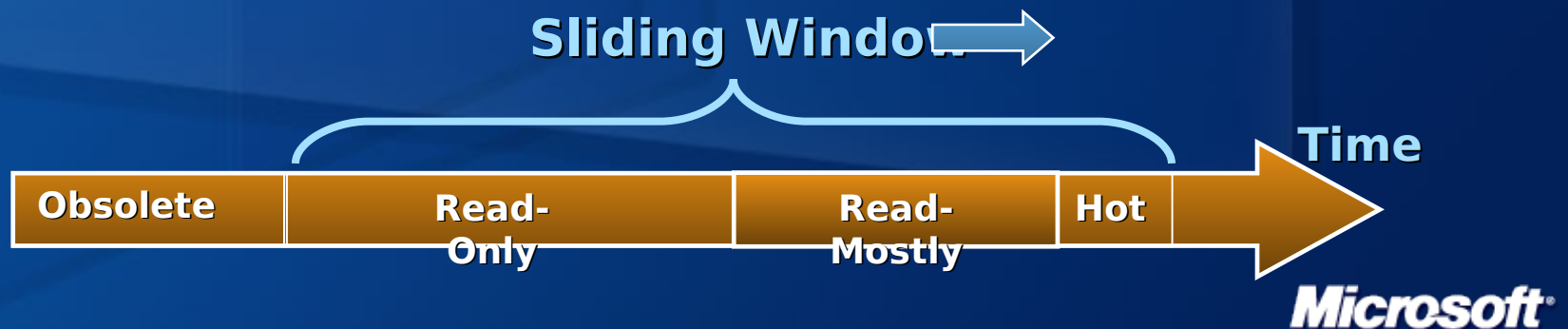
- Isolated hardware failure
 - Only damaged filegroup offline
 - Recovery can include restoring read-only filegroups to their current state without rolling forward transaction logs
- Human error
 - Can take just the damaged filegroup offline
 - If read-only filegroup then only need to recover damaged portion of database—while remainder of database is online
 - Can selectively recover data from database snapshot to manually review/merge lost data (still potentially error prone but easy/FAST solution)

How is This Possible?

- Fine grain operations are based on “partitioning” datasets for VLDB
- Partitioning in this sense does not require the SQL Server 2005 Partitioned Tables feature however, this feature significantly benefits from these capabilities
- Partitioning for fine grain operations just means strategically placing objects within filegroups to recover the prioritized combination at time of disaster
- Strategies...

Date/time-Based

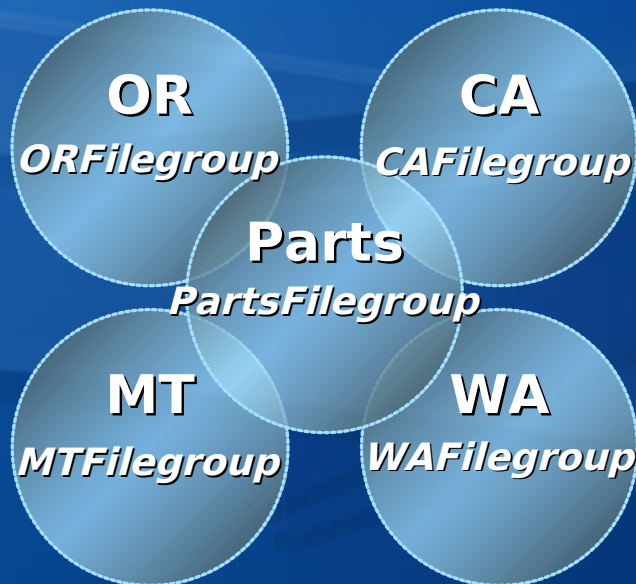
- Time-based data placement
 - Structures designed for sliding window scenario
 - Tables created and data flows on regular/consistent basis – weekly, monthly, yearly, etc.
 - Data may be archived/removed to keep only “current” timeframe – year, two years, etc.
 - Uses SQL Server 2000 Partitioned Views or SQL Server 2005 Partitioned Tables defining “ranges” using date-based criteria



Related-Object Groupings

Related-object groups = List-based or functional

- Regionally based with some shared components
- Functionally based – could use separate tables OR Partitioned Tables using a list-based partition function



Region-specific Data:

- Customers, Sales, SRs are found within the region-specific filegroup
- If CAFilegroup is damaged, customers in Oregon, Washington and Montana are not affected...
- However, damage to “Parts” would mean downtime

Database Structures

Partitioning basics

- Database has at least two files – ALWAYS
 - Data file
 - First data file is the “Primary” data file and stores system tables critical to this database’s accessibility
 - A database will NOT remain available if this is damaged!
 - Critical to isolate (from other data – in a VLDB), create and locate on redundant array
 - Log file
 - Where changes are stored until backed up (unless in Simple recovery model = truncate log on checkpoint)
 - Also critical to database availability, a database will NOT remain available if this is damaged AND you can have data loss if/when this portion of the database is damaged
 - Critical to isolate and make redundant

Database Structures

Partitioning basics (cont'd)

- Additional/Secondary non-Primary Data Files:
 - Basis for “partitioning” data
 - Exist as filegroup(s) and are available as Filegroups
 - A file can ONLY be a member of one filegroup
 - Once added to the database, the filegroup CANNOT be changed (however, the number of files, size and location can and the file can always be removed from the filegroup)
 - Contain user-defined data (tables/indexes) strategically created/placed on one or more filegroups
 - Contain complete objects – when the object has not been partitioned (an object CANNOT exist in multiple files in multiple filegroups unless partitioned)
 - Contain a partition of a partitioned object

Creating Objects on Filegroups

- Objects = Tables/Indexes can be created
 - A Filegroup
 - Can contain one or more files
 - Data is proportionally filled among the files in the filegroup
 - A Partition Scheme
 - Can contain one or more filegroups
 - Data is placed into the appropriate filegroup based on a partition function

CREATE TABLE Customers
(column datatype nullability, ...)
ON FILEGROUP

CREATE TABLE Sales
(column datatype nullability, ...)
ON PartitionScheme

Proportional Fill vs.



Space allocated

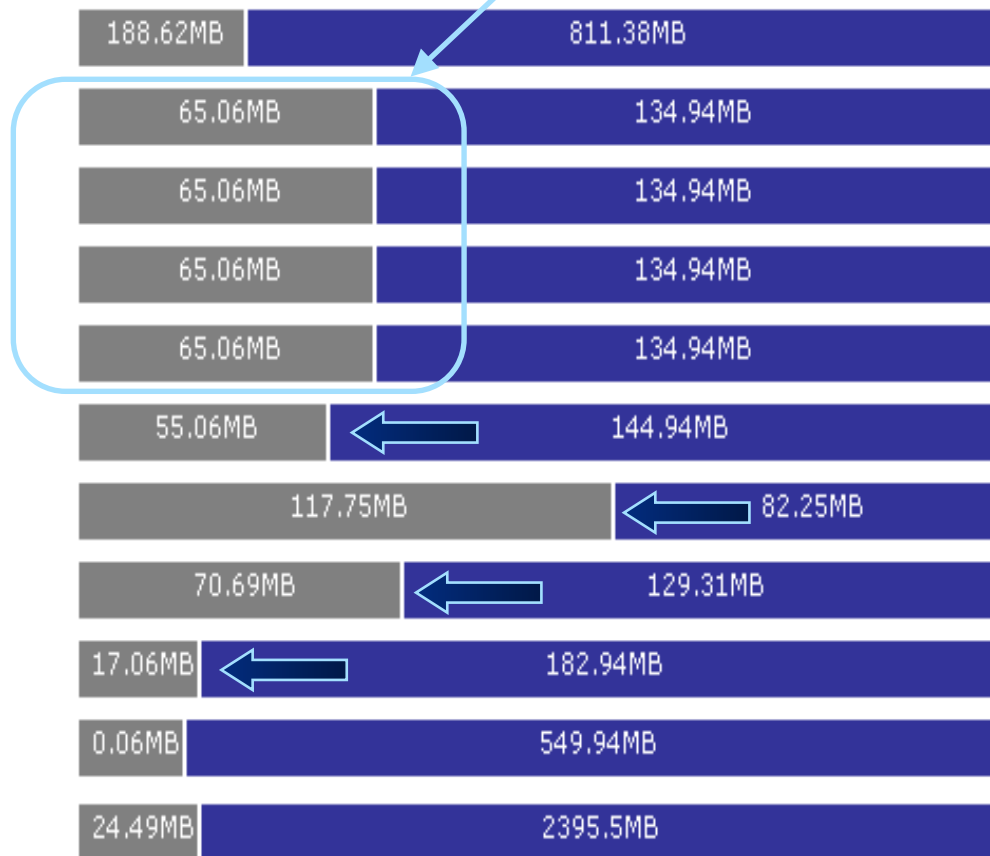
Data:

CreditPartitionedDataPrimary	1000MB
CreditPartitionedSFGFile1	200MB
CreditPartitionedSFGFile2	200MB
CreditPartitionedSFGFile3	200MB
CreditPartitionedSFGFile4	200MB
CreditPartitionedFFGFile1	200MB
CreditPartitionedFFGFile2	200MB
CreditPartitionedFFGFile3	200MB
CreditPartitionedFFGFile4	200MB
ChargesFY04Staging	550MB
Transaction log space:	2419.99MB

Single
Table
Single
FG

Partitioned
Table
Four
FGs

Proportionally filled



Total

Used

Free

Key Differences

- Single Filegroup is easier to create/administer
- CAN perform file/filegroup backups however, no guarantee of where data lives so all files/filegroups must be backed up more frequently vs. frequently backing up ONLY the active partition
- If a file (within a filegroup) becomes damaged the ENTIRE filegroup must be taken OFFLINE
- Cannot manipulate data except at the table level – no concept of data separation or partitions
- Partitioned Table is ORDERS OF MAGNITUDE faster on Rolling Range/Sliding Window operations

Proportional Fill vs.



Space allocated

Data:

CreditPartitionedDataPrimary	1000MB
CreditPartitionedSFGFile1	200MB
CreditPartitionedSFGFile2	200MB
CreditPartitionedSFGFile3	200MB
CreditPartitionedSFGFile4	200MB
CreditPartitionedFFGFile1	200MB
CreditPartitionedFFGFile2	200MB
CreditPartitionedFFGFile3	200MB
CreditPartitionedFFGFile4	200MB
ChargesFY04Staging	550MB
Transaction log space:	2419.99MB

Single
Table
Single
FG

Partitioned
Table
Four
FGs

Proportionally filled



Total

Used

Free

Partitioned Tables and Indexes

Types and implementation

- Types of Partitioning = “Range”
 - Date ranges = defined through boundary cases
 - Does NOT need hard-coded values, each boundary can be based on a function(s) calculated at PF creation
 - Create “list” partitions with no real “ranges” of data
- Implementing Partitioned Tables and Indexes
 - Filegroups/Files
 - Partition Function (logical boundaries)
 - Partition Scheme (interval mapping to physical)
 - Partitioned Table (table targeted at p scheme, specified column goes through p function)
 - Partitioned Index

Range Partitioned Tables

- Step 1: Create Filegroups
- Step 2: Create Files in Filegroups
- *Step 3: Create Partition Function (PF) to define the logical placement of data
- *Step 4: Create Partition Scheme (PS) uses PF and Filegroups to define physical placement of data
- Step 5: Create Table(s)/Index(es) on PS
- Step 6: Add data to tables – SQL Server redirects data and queries to appropriate partition

demo

SQL Server 2005 Partitioned Tables

Reviewing the creation and placement of
data in a Partitioned Table scenario...on
USB keys

(should not be attempted in production ☺)

Benefits of Partitioning

- Speed in managing sliding window
 - ↳ Partition manipulation outside of active table
- Piecemeal backup
 - ↳ Backup active components more frequently, inactive less frequently
- Availability
 - ↳ If a filegroup becomes unavailable the other data can still be accessed and recovery can occur concurrently

demo

SQL Server 2005 Partial Database Availability Accessing a Damaged Database

Damaged Partition

- Does not render the database unavailable – only the damaged filegroup is unavailable
- Does not render the partitioned view OR the partitioned table unavailable – only the damaged data is unavailable
- Can proceed with recovery – also ONLINE!
- May have minimal impact to downtime as files are brought offline and/or when file is restored
- Users can access the database during restore

demo

SQL Server 2005 Piecemeal Online Restore Repairing a Damaged Database ONLINE

Recovering From...

- Hardware
 - Is more defined – damage is obvious and likely to be known immediately
 - Secondary damage significantly limited
- Human error
 - Might not be well understood
 - Damage may not be obvious and full extent of damage may be difficult
 - Secondary damage is possible

Preventative Techniques

- DDL Triggers – synchronous and can trap/rollback unwanted DDL
- Event Notifications – asynchronous and can allow auditing
- No direct user modifications to base tables, limit access through more restrictive permissions to only views, stored procedure
- Agent Alerts based on Error Severity (might be better for hardware failures)

demo

SQL Server 2005 DDL Trigger to Prevent All DDL

Recovery Techniques

- Recover entire database to earlier point in time
 - Pro: Data integrity guaranteed
 - Con: Time-consuming and possibly significant amount of secondary data loss
- Recover part of database to earlier point in time in secondary location and then manually merge
 - Pro: Can roll-forward to any point in time given logs exist for recovery
 - Con: Time-consuming

Recovery Techniques

- Read data from earlier point in time image created by database snapshot and then manually merge
 - Pro: Immediately available, no restore time only select and then insert...select
 - Con: Cannot roll forward beyond point in time at which database snapshot taken
- **TIP:** Create snapshots in a rolling fashion with some consistency and frequency and always keep 1 or 2 online

Creating a Database Snapshot Syntax

- Creating a database snapshot:

```
CREATE DATABASE AdventureWorksDWSnap  
ON (NAME = AdventureWorksDW_Data,  
     FILENAME = 'C:\AWDW.snp')  
AS SNAPSHOT OF AdventureWorksDW
```

Creating a database snapshot creates files, but of what size?

- Dropping a database snapshot:

```
DROP DATABASE AdventureWorksDWSnap
```

Once the snapshot is deleted then all of the NTFS sparse files will also be deleted

demo

SQL Server 2005 Database Snapshot Creating a Database Snapshot

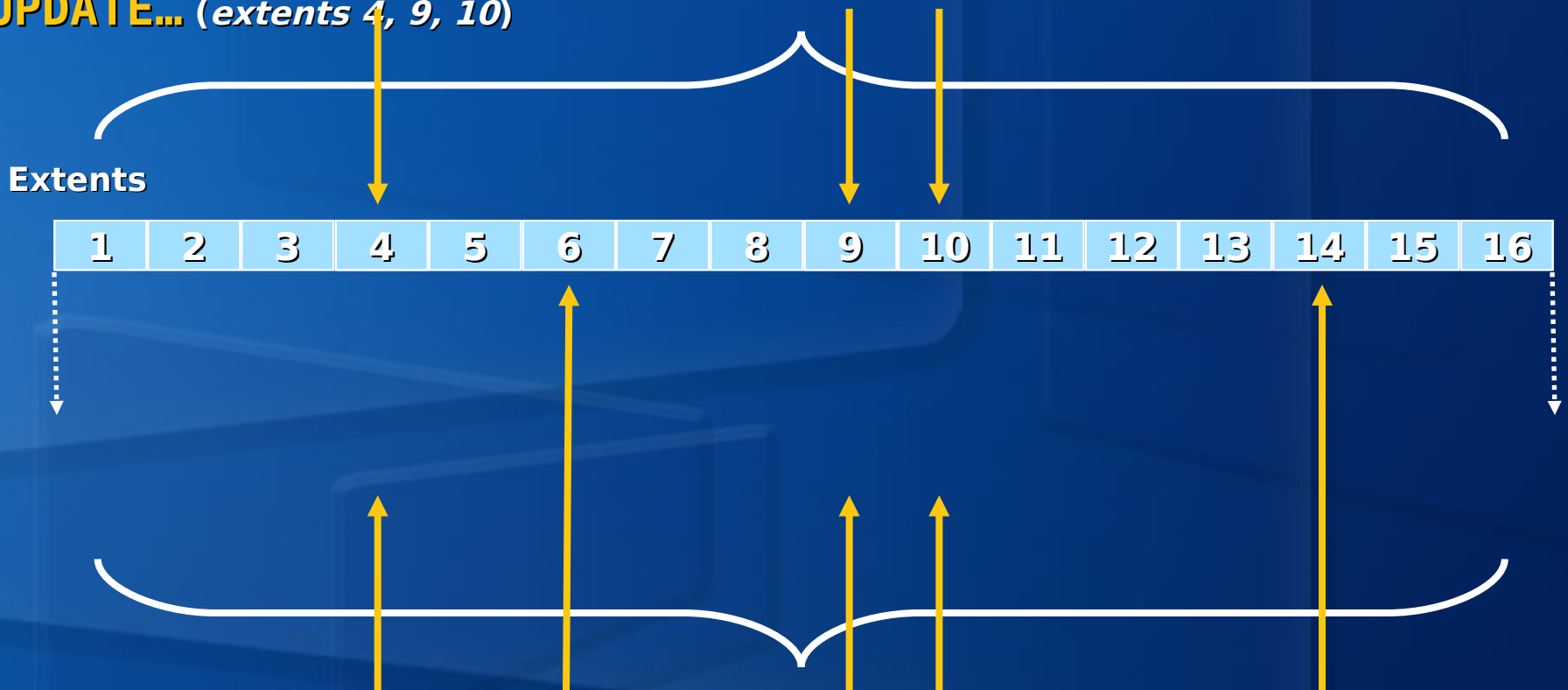
DB Snapshot - How it Really Works

CREATE DATABASE CreditSS1 (...) AS SNAPSHOT OF Credit

USE Credit

UPDATE... (extents 4, 9, 10)

Credit - Database



CreditSS1 - Read-Only Database Snapshot

USE CreditSS1

SELECT... (extents 4, 6, 9, 10, 14)

Creating a Database Snapshot Requirements

- CREATE DATABASE permission required
- Database creation is VERY similar to a database but includes only DATA files with location (no other parameters)
- Resulting Database Snapshot is read-only
- Absolute NO changes are allowed in the database snapshot
 - Files cannot be dropped
 - Permissions cannot be changed
 - Indexes cannot be added/rebuilt, etc.
- Database snapshots cannot be created for the model, master, and temp databases
- Come up with naming convention for snapshot files as well as database name (time of day?)

Creating a Database Snapshot

Secondary considerations

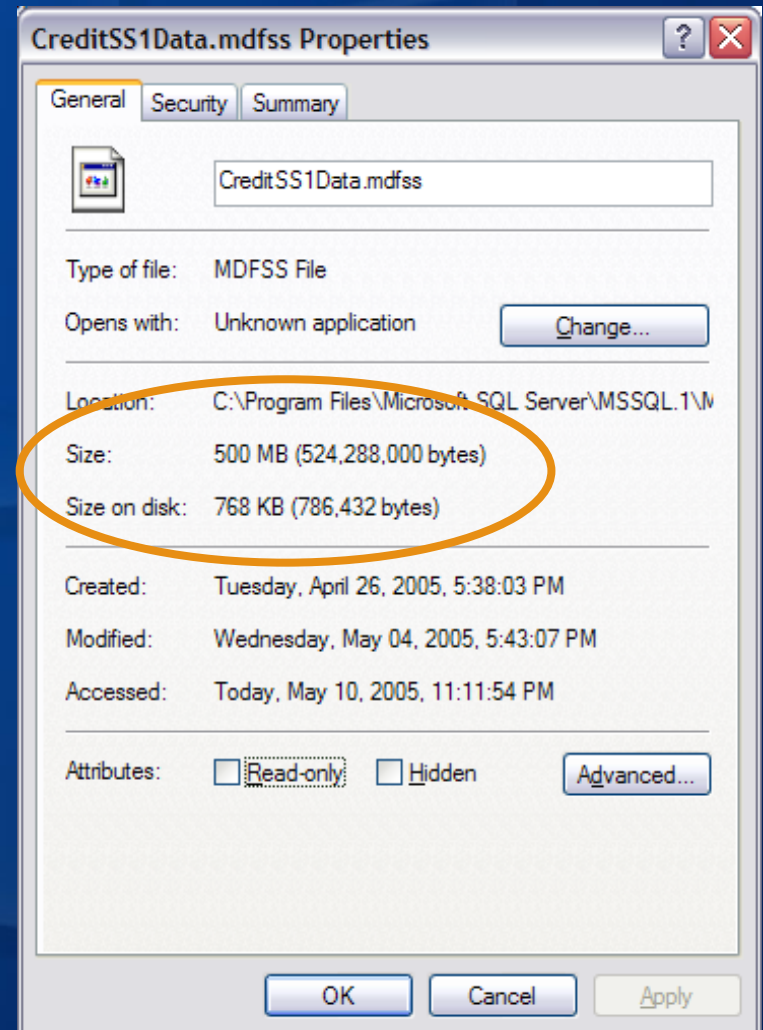
- DB snapshot and source database must be on the same instance
- If any database snapshots exist on a source database, the source database cannot be dropped or restored (must delete all of the database snapshots first)
- Backups are not allowed on the snapshot, however, source database backups are unaffected by database snapshots

Database Snapshot Technology

- Extremely space efficient
- Does **not** require a complete copy of the data
 - Shares unchanged pages of the database
 - Requires storage only for changed pages
- Uses a “copy-on-write” mechanism via NTFS sparse files
- Database Snapshot(s) may affect performance on the base database. However, only on the first change.

How Much Space Used?

- Use Windows Explorer to right click on the NTFS sparse, select Properties
- Look at the “Size” and you will see it is the same size of the source database file
- However, look at the “Size on Disk” and you will see how much space is REALLY used



How Much Space Used?

- Internally you can use catalog views and DMVs (Dynamic Management Views) to see the actual size

```
SELECT db_name(v.database_id) AS [DatabaseName]
      , v.file_id AS [File ID]
      , CASE WHEN is_sparse = 0 THEN 'No'
            ELSE 'Yes' END AS [Sparse File]
      , f.[name] AS [File Name]
      , size_on_disk_bytes/1024 AS [Size (KB)]
      , f.physical_name AS [Physical File Name]
FROM sys.dm_io_virtual_file_stats
      (db_id('dbname'), -1) AS v
  JOIN sys.master_files AS f
      ON v.file_id = f.file_id
WHERE f.database_id = v.database_id
AND f.database_id = db_id('dbname')
```

demo

SQL Server 2005

Database Snapshot

**Viewing NTFS Sparse Files space,
Accessing Data, Creating a second
database snapshot, Reverting**

Using Database Snapshot Data

Query and recover damaged data and/or use it for a database level rollback

- Responding to human error
 - Use data from snapshot and manually merge in – but you cannot roll forward!
 - Only gives you access to a specific point in time
 - Must be pro-actively created!
- Reverting the ENTIRE database to an earlier state, great for testing!
 - Create a database snapshot
 - Perform testing/performance operations
 - Revert to snapshot
 - Re-run different tests

Using Database Snapshot Data

Query and recover damaged data and/or use it for a database level rollback

- Use 3 part naming

```
INSERT DamagedDB.schema.object  
SELECT data  
FROM DBSnapshot.schema.object  
WHERE ...
```

- Revert to a snapshot

```
RESTORE DATABASE Credit  
FROM database_snapshot = 'Creditss1'
```

- Note: Reverting to a snapshot does NOT drop the snapshot

Recovering from Human Error

Minimizing downtime and/or data loss - making the choice

- Use point in time data from database snapshot and manually merge in:
 - Excellent when the table is relatively static
 - Poor choice for an extremely volatile table – cannot roll forward with log backups...
- Restore backups to alternate location
 - Restore database in full or partial and roll forward as far as possible
 - Consider using RESTORE with STANDBY to simultaneously investigate while restoring

See **SQL Server Magazine Article: InstantDoc #39657**

Resources

- Whitepaper: SQL Server 2005 Snapshot Isolation
On MSDN, link on home page of www.SQLskills.com
- Whitepaper: SQL Server 2005 Partitioned Tables
On MSDN, link on home page of www.SQLskills.com
- Whitepaper: Strategies for Partitioning Relational Data Warehouses in Microsoft SQL Server
 - <http://www.microsoft.com/technet/prodtechnol/sql/2005/spdw.mspx>
- Whitepaper: Using Partitions in a Microsoft SQL Server 2000 Data Warehouse
 - <http://msdn.microsoft.com/library/en-us/dnsq12k/html/partitionsindw.asp?frame=true>

Resources

- Check out www.SQLskills.com for information about upcoming **SQL Immersion** events, useful links and event scripts.
- Read my blog:
<http://www.SQLskills.com/Blogs/Kimberly/>
- Subscribe to SQLskills:
<http://www.SQLskills.com/login.aspx>
- MPress: *SQL Server 2000 High Availability*
Authors: Allan Hirt with Cathan Cook, Kimberly L. Tripp and Frank McBath
ISBN: 0-7356-1920-4
On the SQLskills.com homepage can download a sample chapter



Community Resources

- Microsoft Community Resources
<http://www.microsoft.com/communities/default.mspix>
- Non-Microsoft Community Resources
<http://www.microsoft.com/communities/related/default.mspix>
- Newsgroups
Converse online with Microsoft Newsgroups, including Worldwide
<http://www.microsoft.com/communities/newsgroups/default.mspix>
- User Groups
Meet and learn with your peers
<http://www.microsoft.com/communities/usergroups/default.mspix>
- Attend a free chat
<http://www.microsoft.com/communities/chats/default.mspix>
- Attend a free web cast
<http://www.microsoft.com/usa/webcasts/default.asp>
- Most Valuable Professional (MVP)
<http://mvp.support.microsoft.com/>

Microsoft Learning Resources:

Get ready for Microsoft Visual Studio 2005 and Microsoft SQL Server 2005 with free Assessments and E-Learning, and a chance to win a laptop, GPS, and more

- Click here to access free Microsoft Learning Assessments
<http://www.microsoft.com/learning/assessment/ind/default.asp>
- To access free Microsoft Learning E-Learning visit
<http://www.microsoft.com/learning/access> and reference the promotional code **9185-TECHED-6650**.

Special offers on Microsoft Certification from Microsoft Learning

- Click here to take advantage of these special offers:
<http://www.microsoft.com/learning/mcp/>



All attendees who submit a session feedback form within 120 minutes after the session ends will have the chance to win an **MPx220** With **Windows Mobile™ Software.**

Added Bonus!! This year you will also have the chance to win one of eight Xbox 360s! We will be giving away two a day – one for the morning sessions, and one for the afternoon sessions.

Good luck!



The Microsoft logo is displayed in a bold, italicized, white sans-serif font. It features a registered trademark symbol (®) at the end. The logo is centered horizontally and has a subtle drop shadow effect against the blue background.

Microsoft[®]

© 2005 Microsoft Corporation. All rights reserved. This presentation is for
informational purposes only.
MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.